

520.43557X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: K. HATASAKI

Serial No.:

Filed: March 2, 2004

For: System-Updating Method And Computer System
Adopting The Method

Group:

Examiner:

LETTER CLAIMING RIGHT OF PRIORITY

Mail Stop: New Appln.
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

March 2, 2004

Sir:

Under the provisions of 35 USC 119, applicant hereby claims the right of priority based on Japanese Patent Application No. 2003-339487, filed September 30, 2003.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/jla
(703) 312-6600

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 9 月 3 0 日
Date of Application:

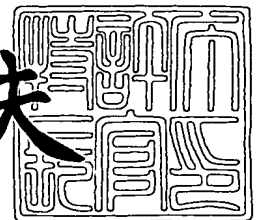
出 願 番 号 特 願 2 0 0 3 - 3 3 9 4 8 7
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 3 3 9 4 8 7]

出 願 人 株 式 会 社 日 立 製 作 所
Applicant(s):

2 0 0 4 年 2 月 1 2 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 NT03P0650
【提出日】 平成15年 9月30日
【あて先】 特許庁長官 殿
【国際特許分類】 G09F 9/06
G09F 15/00

【発明者】
【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所
中央研究所内
【氏名】 畑▲崎▼ 恵介

【特許出願人】
【識別番号】 000005108
【氏名又は名称】 株式会社日立製作所

【代理人】
【識別番号】 100068504
【弁理士】
【氏名又は名称】 小川 勝男
【電話番号】 03-3661-0071

【選任した代理人】
【識別番号】 100086656
【弁理士】
【氏名又は名称】 田中 恭助
【電話番号】 03-3661-0071

【選任した代理人】
【識別番号】 100094352
【弁理士】
【氏名又は名称】 佐々木 孝
【電話番号】 03-3661-0071

【手数料の表示】
【予納台帳番号】 081423
【納付金額】 21,000円

【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1

【書類名】 特許請求の範囲**【請求項 1】**

ユーザ計算機システムにインストールされているソフトウェアを更新する方法において、

前記ユーザ計算機システムにインストールされているソフトウェア情報、ハードウェア情報を含むユーザシステム情報を取得するステップと、

取得したユーザシステム情報を基に、システム動作をテストするためのテスト環境を構築するステップと、

前記テスト環境においてソフトウェア更新パッチを用いてソフトウェアの更新をおこなうステップと、

前記テスト環境において前記更新されたソフトウェアによりシステムが正常に動作するか否かを確認するステップと、

システムの正常動作が確認された場合に、前記ソフトウェア更新パッチを前記ユーザ計算機システムに送付して、前記ユーザ計算機システムのソフトウェアを更新するステップとを有することを特徴とするシステム更新方法。

【請求項 2】

前記ユーザ計算機システムは、

前記ユーザシステム情報を監視する手段と、

ユーザ計算機システムにインストールされているソフトウェアを更新する手段とを有し、

前記ユーザ計算機システムは、前記ソフトウェア更新パッチを取得して、前記ユーザ計算機システムの更新対象となるソフトウェアに係るプログラムを停止させることなく更新することを特徴とする請求項 1 記載のシステム更新方法。

【請求項 3】

前記ユーザ計算機システムは、

前記テスト環境におけるシステム動作のテスト結果を取得し、

ユーザに前記テスト結果を表示して、

ユーザ計算機システムのソフトウェア更新の是非を、前記ユーザに問い合わせることを特徴とする請求項 1 記載のシステム更新方法。

【請求項 4】

ユーザ計算機システムにインストールされ、ベンダ計算機システムから提供を受けるソフトウェアを更新するシステム更新方法において、

ユーザシステム情報を前記ベンダ計算機システムに送信するステップと、

前記ベンダ計算機システムにおいて、受信した前記ユーザシステム情報に基づき適切なソフトウェア更新パッチを選択して、前記ユーザ計算機システムにソフトウェア更新パッチを送信するステップと、

前記ベンダ計算機システムから受信したソフトウェア更新パッチを基にテスト環境を構築するステップと、

前記テスト環境を利用し前記ソフトウェア更新パッチを評価して評価結果を求めるステップと、

前記評価結果に従って、前記評価結果が良いときには、前記ソフトウェア更新パッチを利用してユーザ計算機システムにインストールされているソフトウェアを更新するステップと、

前記評価結果に従って、前記評価結果が悪いときには、前記ベンダ計算機システムに、前記評価結果を送付して、前記ソフトウェア更新パッチの再送付を促すステップとを有することを特徴とするシステム更新方法。

【請求項 5】

ユーザ計算機システムにソフトウェアを提供するベンダ計算機システムにおいて、

前記ユーザ計算機システムのユーザシステム情報を受信する手段を有するユーザシステム管理部と、

前記ユーザシステム情報とソフトウェア更新パッチとにより、ユーザ計算機システムのテスト環境を構築する手段と、

前記テスト環境を利用して、ユーザ計算機システムに適合したソフトウェア更新パッチを評価する手段と、

前記ソフトウェア更新パッチを前記ユーザ計算機システムに送付する手段とを有することを特徴とするベンダ計算機システム。

【請求項 6】

前記ベンダ計算機システムを構成する実計算機を、複数の論理計算機に論理的に分割する手段と、前記論理計算機に前記テスト環境を構築する手段とを有することを特徴とする請求項 5 記載のベンダ計算機システム。

【請求項 7】

前記ソフトウェア更新パッチをパッチデータベースに記録する手段と、

前記ユーザ計算機システムからユーザシステム更新要求を取得し、前記ユーザシステム更新要求に基づき前記パッチデータベースから適切なソフトウェア更新パッチを取得する手段とを有することを特徴とする請求項 5 記載のベンダ計算機システム。

【請求項 8】

前記ユーザ計算機システムから受信した前記ユーザシステム情報をユーザシステム情報データベースに記録する手段と、

前記ユーザシステム情報を前記システム情報データベースから取得する手段とを有することを特徴とする請求項 5 記載のベンダ計算機システム。

【請求項 9】

ベンダ計算機システムからソフトウェアの提供を受けるユーザ計算機システムにおいて、

ユーザシステム情報を前記ベンダ計算機システムに送信する手段と、

前記ベンダ計算機システムからソフトウェア更新パッチを受信する手段と、

前記ベンダ計算機システムから受信したソフトウェア更新パッチと自らのユーザシステム情報を基にテスト環境を構築する手段と、

前記テスト環境を利用して前記ソフトウェア更新パッチを評価して評価結果を求める手段と、前記評価結果をベンダに送付する手段とを備えることを特徴とするユーザ計算機システム。

【請求項 10】

前記ユーザ計算機システムを構成する実計算機を、複数の論理計算機に論理的に分割する手段と、前記論理計算機に前記テスト環境を構築する手段とを有することを特徴とする請求項 9 記載のユーザ計算機システム。

【書類名】明細書**【発明の名称】** システム更新方法、および、それを実行するための計算機システム**【技術分野】****【0001】**

本発明は、システム更新方法、および、それを実行するための計算機システムに係り、特に、ユーザシステムのソフトウェアを更新することの多いオープンシステムに用いて好適であって、ユーザシステムにおけるソフトウェア更新作業を容易におこなうことができ、しかも、ソフトウェア更新後のユーザシステムの信頼性とサービスレベルを保証することのできるシステム更新方法、および、それを実行するための計算機システムに関する。

【背景技術】**【0002】**

ユーザの計算機システム（以下、単に「ユーザシステム」という）を更新する方法としては、従来では、CD-ROMなどの記憶媒体からソフトウェアをインストールする方法が殆どであった。

【0003】

しかしながら、近年では、インターネットの普及などにより、ネットワークを介してソフトウェアを提供するベンダ計算機システム（以下、単に「ベンダシステム」という）からソフトウェアをダウンロードして、ユーザシステムにインストールする方法も広くおこなわれるようになってきている。

【0004】

このネットワークによるソフトウェアインストール方法では、例えば、ユーザシステム側でユーザシステムのソフトウェア構成情報を生成し、ユーザシステムがベンダシステムのパッチ管理サーバからの定期的にソフトウェア更新情報をネットワークを通じて受信する。そして、ユーザシステムのソフトウェア構成情報と受信したソフトウェア更新情報とを比較して、更新すべきソフトウェアを調査し、調査結果から更新すべきソフトウェア一覧を表示する。ユーザがそれを選択して、更新を要求すると、ベンダシステムは、ソフトウェアの更新パッチをユーザシステムにネットワークを通じて送信する。ユーザシステムでは、その更新パッチを更新するソフトウェアに適用して、ソフトウェアを更新する。ここで、「パッチ」とは、プログラムの修正をおこなうための差分ファイルである。

【0005】

例えば、以下の特許文献1には、サーバ側にパッチデータベースを構築しておき、クライアントは、ネットワークを介してパッチ情報を受信して、自らのコンピュータシステムに適用する例が開示されている。

【0006】

また、ミッションクリティカルなユーザシステムを更新する場合には、ベンダ側のシステム管理者がユーザシステムにソフトウェア更新パッチを持ちこみ、更新したいソフトウェアに対して稼動中に前記ソフトウェア更新パッチを適用して、ユーザシステムやソフトウェアの再起動をせずに更新することで、可用性を低下させずに更新する方法も存在する。

【0007】

さらに、ユーザシステムを構築したベンダがユーザシステムと同じシステムをベンダ側で準備して、ユーザシステムの更新をベンダ側でテストする方法がある。

【0008】

【特許文献1】 特開2002-55839号公報

【発明の開示】**【発明が解決しようとする課題】****【0009】**

ネットワークによるユーザシステムのソフトウェアの更新は、記録媒体による方法に比べて、手軽にでき、しかも、最新のソフトウェアを利用できるメリットがある。

【0010】

しかしながら、従来のネットワークによるソフトウェア更新方法では、ユーザシステムの更新後に、更新のよって動作が変化したユーザシステムが障害を発生して、システム停止や再起動などの発生により可用性が低下する場合や、ユーザシステムの更新の結果、ユーザシステムに要求されるサービスレベルに達しない場合や、ユーザシステムの更新によってユーザが所望の機能を実行できなくなるなどなどの問題が発生する可能性がある。

【0011】

特に、セキュリティにおける脆弱性の発覚やバグの発生など、重大な原因によってソフトウェアを更新したい場合には、上記の問題は大変な損害を発生する可能性が高い。また、24時間365日稼動していることが必要なシステムを更新する場合には、確実にソフトウェアの更新ができなければならない。

【0012】

また、ベンダが作成した一意なソフトウェア更新パッチは、ユーザシステムのシステム構成によっては必ずしも正しく動作しない可能性がある。特にオープンソースなどのソフトウェアを利用するユーザシステムでは、ユーザ自身によってソフトウェアのソースコードの書き換えが可能であるため、ユーザシステムはユーザ固有の構成となっている可能性がたかい。このために、ユーザシステムにベンダが作成した一意なパッチが適応できないことが多い。

【0013】

本発明は、上記問題点を解決するためになされたもので、その目的は、ネットワークを介して、ソフトウェアを更新する方法において、ユーザシステムにおけるソフトウェア更新作業を容易におこなうことができ、しかも、ソフトウェア更新後のユーザシステムの信頼性とサービスレベルを保証することのできるシステム更新方法を提供することにある。

【課題を解決するための手段】**【0014】**

上記の課題を解決するため、本発明に係るシステム更新方法は、ベンダシステムで、ユーザシステムにインストールされているソフトウェア情報やハードウェア情報を含むユーザシステム情報を取得する。そして取得したユーザシステム情報を基にテスト環境を構築して、そのテスト環境においてソフトウェア更新パッチを適用したシステムの動作を検証し、テスト環境においてシステムが正常に動作するかを確認する。そして、正常動作が確認された場合に、ソフトウェア更新パッチをユーザシステムに送信し、ユーザシステムでは、そのソフトウェア更新パッチを適用して、自らのシステムを更新する。

【0015】

また、テスト環境をユーザシステム側に構築し、最適なソフトウェア更新パッチをベンダシステム側から受信する仕組みにしてもよい。

【0016】

これにより、ユーザシステムと同じ環境のテスト環境でシステム動作が確認されたソフトウェア更新パッチのみがユーザに配布されることになり、ユーザシステムでのソフトウェアの更新によるトラブルをなくすることができる。

【0017】

すなわち、ユーザシステムの更新に利用するソフトウェア更新パッチは、ユーザシステムと同等のテスト環境によってテストに成功したものであり、ユーザシステムの更新時にシステムがクラッシュすることなく、確実に更新することができる。また、更新後の機能や性能についてもテストが実施できるため、ユーザの要求するサービスレベルを確実に達成し、かつ、必要な機能を確実に追加することができる。

【0018】

また、ユーザシステム情報をベンダが得て、ユーザシステムと同等のテスト環境を利用してテストを実施するため、オープンソースソフトウェアをユーザシステムにインストールして利用しているユーザが、オープンソースソフトウェアを自身で書き換えた場合や、ユーザ固有の環境でしか発生しない問題が存在する場合であっても、ユーザシステムに適合したソフトウェア更新パッチを作成することができる。そのために、このようなユーザ

システム固有の問題にも対処することができる。

【発明の効果】

【0019】

本発明によれば、ネットワークを介して、ソフトウェアを更新する方法において、ユーザシステムにおけるソフトウェア更新作業を容易におこなうことができ、しかも、ソフトウェア更新後のユーザシステムの信頼性とサービスレベルを保証することのできるシステム更新方法を提供することができる。

【発明を実施するための最良の形態】

【0020】

以下、本発明に係る各実施形態を、図1ないし図16を用いて説明する。

【0021】

〔実施形態1〕

以下、本発明に係る第一の実施形態を、図1ないし図15を用いて説明する。

【0022】

先ず、図1を用いて本発明に係る第一の実施形態のシステム更新方法を実現するためのシステム構成を説明する。

図1は、本発明に係る第一の実施形態のシステム更新方法を実現するためのシステム構成図である。

【0023】

本実施形態の計算機システムは、図1に示されるようにユーザがプログラムを実行するためのユーザシステム110と、ユーザにソフトウェアを提供するベンダベンダシステム130がネットワーク140により接続された構成をしている。

【0024】

ユーザシステム110は、アプリケーション121と、ミドルウェア122と、OS123と、システム更新部111とから構成されるシステムである。

【0025】

システム更新部111は、ユーザシステム110のユーザシステム情報101を収集しネットワーク140などの遠隔通信手段によってベンダシステム130へユーザシステム情報101を送信する機能と、アプリケーション121、ミドルウェア122、OS123のコンポーネントを更新する機能を有する。

ここで、OS123には、ユーザシステム110に組み込まれているライブラリや、デバイスドライバや、モジュールなどのソフトウェアも含まれている。

【0026】

一方のベンダシステム130は、ユーザシステム管理部131と、パッチテスト部132と、パッチ送信部134と、テスト環境201とから構成されるシステムである。

【0027】

パッチテスト部132は、テスト環境構築部133を含み、修正ソフトウェア更新パッチ102を読み込み、そのソフトウェアによるシステムのテスト環境を構築して、システム動作を確認する機能を有する。そして、テストに成功するようにソフトウェア更新パッチを修正し、修正後ソフトウェア更新パッチ103を生成する。

【0028】

パッチ送信部134は、テストに成功した修正後ソフトウェア更新パッチ103をネットワークにより、ユーザシステム110に送信する。

【0029】

次に、図2ないし図7を用いて本発明に係る第一の実施形態のシステム更新方法の手順を説明する。

図2は、本発明に係る第一の実施形態のシステム更新方法の手順の概要を示すシーケンス図である。

図3は、ユーザシステム情報の一例を示す図である。

図4は、被テストソフトウェア更新パッチの一例を示す図である。

図5は、ユーザシステムの表示装置で、更新パッチの種別を選択する場面を示す模式図である。

図6は、ユーザシステムの表示装置で、更新テスト結果を確認し、更新を実行する場面を示す模式図である。

図7は、ユーザシステムのシステム更新の一形態を示す模式図である。

【0030】

以下、図2を追いながら他の図面を随時参照して、本実施形態のシステム更新方法の手順を説明することにする。

【0031】

図2に示されるように、ユーザシステム102は、アプリケーション121、ミドルウェア122、OS123のレイヤを持ち、アプリケーション121は、更新前のコンポーネントa1, a2, ..., ax、ミドルウェア122は、更新前のコンポーネントm1, m2, ..., mx、OS123は、更新前のコンポーネントk1, k2, ..., kxから成ることにする。

【0032】

そして、各レイヤのコンポーネントは、テスト環境201では、置き換えられる。本実施形態の説明では、更新後のコンポーネントは、それぞれのアルファベットの大文字で表すことにする。

【0033】

先ず、ユーザシステム110が、システム更新を要求し(S11)、ベンダシステムは、ユーザシステムからのシステム更新の要求を受信する(S21)。

【0034】

このS11のステップで、システム更新要求をするときには、例えば、ユーザシステム110では、図6に示されるように、ブラウザなどを利用して、リクエスト送信インタフェース710をユーザシステムの表示装置701に表示し、更新に利用できるパッチをパッチリスト711に表示する。そして、ユーザは、パッチリスト711から選択ボタン712をチェックして選択し、リクエスト送信ボタン713をマウスなどのポインティングデバイスにより、クリックしてベンダに送信する。

【0035】

次に、ユーザシステム110のシステム更新部111が、ユーザシステム情報101を取得して、取得したユーザシステム情報101をユーザシステム管理部131に送信する(S12)。システム更新部111がユーザシステム情報101を取得する方法としては、エージェントプログラムをユーザシステム110上で動作させてユーザのユーザシステム110に対する変更を常に監視する方法や、ユーザ自身がユーザシステム情報101をベンダに通知するためのユーザインタフェースを用意する方法がある。また、S12のステップの処理は、必ずしもS11の処理の後に実行せずに、ユーザによるユーザシステム110の更新の度に実行しても良い。

【0036】

ここで、ユーザシステム情報101は、図3に示すように、ハードウェア情報310と、ソフトウェア情報320や、ソースコード情報とから成る情報である。

【0037】

ハードウェア情報310は、ユーザシステムのプロセッサの種類や、メモリの容量や、チップセットの種類や、デバイスの種類などからなる情報である。

【0038】

ソフトウェア情報320は、ユーザシステムにインストールされているアプリケーション121のコンポーネントの種類や、ミドルウェア122のコンポーネントの種類や、OS123のコンポーネントの種類などからなる情報である。

【0039】

ソースコード情報330は、アプリケーション121、ミドルウェア122、OS123のソースコードと、それに関連する情報である。ここで、本実施形態で、ユーザシステ

ム情報101にソースコード情報330を含めたのは、ソースコードが公開されるオープンシステムへの適用を意識したものである。

【0040】

次に、ベンダシステム130は、ユーザシステム管理部131からS12の処理で送信されたユーザシステム情報101を受信する(S22)。

【0041】

ベンダシステム130では、パッチテスト部132のテスト環境構築部133が、ステップS22の処理で受信したユーザシステム情報101を受け取り、さらに、ユーザシステム110の更新に必要な被テストソフトウェア更新パッチ102を取得し、ユーザシステム110と同じ環境を持つテスト環境201を構築する(S23)。

【0042】

本実施形態では、ユーザシステム110からユーザシステム情報101を直接受信したが、ベンダシステム130で、ユーザシステム情報データベースを構築して置き、そこから取得するようにしてもよい。

【0043】

また、被テストソフトウェア更新パッチ102は、例えば、S11の処理でユーザが図13のパッチリスト711から選択したパッチである。被テストソフトウェア更新パッチ102の取得方法は、ユーザから入手する場合や、ベンダ自身が作成した場合や、オープンソースコミュニティなどのその他の組織や個人から入手する場合がある。あるいは、ベンダは被テストソフトウェア更新パッチ102が必要になった場合にすぐに取得できるように、ベンダがパッチデータベースを用意して、複数のソフトウェア更新パッチを記録しておき、必要な時に必要なソフトウェア更新パッチをパッチデータベースから取得できるようにする方法もある。

【0044】

被テストソフトウェア更新パッチ102は、ソースコードパッチ410、バイナリコードパッチ420のいずれか一方、または、両方からなるソフトウェア更新パッチ102である。

【0045】

ソースコードパッチ410は、更新の対象となるソフトウェアの更新前のソースコードと更新後のソースコードの差分である。バイナリコードパッチ420は、更新の対象となるソフトウェアの更新前のバイナリコードと更新後のバイナリコードのバイナリ形式の差分である。また、被テストソフトウェア更新パッチ102をテストして作られるテスト成功ソフトウェア更新パッチ103の構成も、これと同じ構成になる。

【0046】

なお、テスト環境構築部133の構成と動作については、後に詳細に説明する。

【0047】

構築されたテスト環境201では、ユーザシステムと同じ環境を構築し、修正前更新パッチによりソフトウェアを更新して、システム動作をテストするための検証テストが実施される(S24)。

【0048】

そして、検証テストに失敗した被テストソフトウェア更新パッチ102を、S24のステップで得たテスト結果を元に修正し、テスト環境構築部133において修正したソフトウェア更新パッチを被テストソフトウェア更新パッチ102の代わりに利用してテスト環境を構築し、再度検証テストを実施する(S25)。このS24とS25のステップの処理をテストが成功するまで繰り返す。

【0049】

また、S25のステップにおいて、S24のステップでの検証テストのテスト結果をユーザに通知し、ユーザに更新の是非を問い合わせる方法もある。

【0050】

例えば、図6に示すようにブラウザなどを利用して、ユーザシステムの表示装置801

に更新情報表示インタフェイス 810 を用意し、S24 のステップのテスト結果である更新テスト結果情報 811 を表示する。そして、ここで、ユーザが更新実行ボタン 812 をクリックすることによって、検証テストに成功したとみなすことにする。

【0051】

ベンダシステム 130 では、次に、パッチ送信部 131 が、S25 のステップの処理で検証テストに成功したテスト成功ソフトウェア更新パッチ 103 をユーザのシステム更新部 111 に送信する (S26)。

【0052】

ユーザシステム 110 側では、ベンダシステム 130 からテスト成功ソフトウェア更新パッチ 103 を受信し、システム更新部 111 がテスト成功ソフトウェア更新パッチ 103 を取得して、テスト成功ソフトウェア更新パッチ 103 をユーザシステム 110 のアプリケーション 121、ミドルウェア 122、OS 123 に適用して、ユーザシステム 110 を更新する (S13)。

【0053】

図の例では、ミドルウェア 122 のコンポーネント m1 が M1 に、OS 123 のコンポーネント k2 が K2 に更新されたことを示している。

【0054】

ユーザシステム 110 の事情が許せば、ユーザシステム 110 を再起動して更新してもよいが、ユーザシステム 110 のプログラムやシステムを再起動せずに更新することができる。

【0055】

例えば、図7では、システム更新部 111 が OS 123 を再起動せずに更新する一例を示している。図7の旧コード C10 は、稼動中の OS 123 のバイナリコードであり、新コード C20 は、テスト成功ソフトウェア更新パッチのバイナリコードパッチ 420 か、あるいは、システム更新部 111 がソースコードパッチ 410 をコンパイルして生成したバイナリコードである。システム更新部 111 は、新コード C20 を OS 123 に挿入して、旧コード C10 の先頭に、新コード C20 への分岐命令 C01 を挿入することにより、OS 123 を更新する。

【0056】

次に、図8ないし図15を用いてテスト環境構築部 133 の詳細について説明する。

図8は、テスト環境構築部 133 の構成を示すブロック図である。

図9は、テスト環境 201 の構成例を示す模式図である。

図10ないし図15は、テスト環境構築部 133 の各部の動作を示すフローチャートである。

【0057】

テスト環境構築部 133 は、図8に示すようにプラットフォーム選択部 510 と、ソフトウェア取得部 520 と、コンパイル部 530 と、インストール部 540 と、パッチ適用部 550 と、検証テスト実行部 560 とから成り、テスト環境 201 を構築する。

【0058】

テスト環境 201 は、異なるハードウェア構成をもつ複数のプラットフォーム 571 を有し、それぞれのプラットフォーム P1, P2, Pn には、インストールされているアプリケーションのコンポーネント a1, a2, ax と、ミドルウェアのコンポーネント m1, m2, mx と、OS のコンポーネント k1, k2, kx が示されている。

【0059】

また、テスト環境 201 は、実際に異なったハードウェア環境を用意して構築する方法のほかに、図9に示されるように、プラットフォーム 571 の実計算機を複数の論理計算機 (Logical Partition) 510 に、論理的分割して、その各々の論理計算機 510 に割り当てる方法もある。ここで、管理計算機 530 は、論理計算機 510 の CPU 数や CPU の動作周波数、メモリ容量の設定や、論理計算機 510 に接続するデバイス 520 を選択する計算機である。

【0060】

テスト環境構築部133は、以下の手順でユーザシステム110からユーザシステム情報101を取り込み、ユーザシステムの同じ環境を持つテスト環境201を構築して、構築したテスト環境を利用して検証テストを実行する。

【0061】

先ず、プラットフォーム選択部510は、図10に示すように、ユーザシステム情報101から、ハードウェア情報310を取得する(S511)。次に、取得したハードウェア情報310を元に、テスト環境570のプラットフォーム571からユーザシステム110と一致するハードウェア構成を持つプラットフォーム571を選択する(S512)。そして、必要であれば選択したプラットフォーム571に論理計算機610を生成し、論理計算機610の一つをユーザシステム110と一致するハードウェア構成となるように、CPU数やCPUの動作周波数、メモリ容量などを設定し、接続するデバイスを選択する(S513)。

【0062】

ここで、ユーザシステム110のハードウェア構成と同等なプラットフォーム571をベンダが入手困難な場合は、ソフトウェアによるエミュレーションにより擬似的にプラットフォーム571を構築する方法もある。プラットフォームに論理計算機を利用せずに、実計算機を用いても良い。また、複数の計算機やデバイスをベンダが用意しておき、テスト環境201を構築する必要がある場合にその一つをテスト環境用のプラットフォーム571として利用し、不要な場合は別の用途に利用する方法もある。

【0063】

次に、ソフトウェア取得部520は、図11に示すように、ユーザシステム情報101から、ソフトウェア情報320を取得する(S521)。次に、取得したソフトウェア情報320を元に、ユーザシステム110にインストールされているアプリケーション121、ミドルウェア122、OS123のコンポーネントをソフトウェアデータベース521から取得する(ステップS522)。

【0064】

ここで取得するソフトウェアのコンポーネントは、実行可能なバイナリ形式であっても、ソースコードであっても良い。

【0065】

次に、コンパイル部530は、図12に示すように、ユーザシステム情報101からソースコード情報330を取得する(S531)。次に、被テストソフトウェア更新パッチ102を取得して、前記ソースコード情報330およびソフトウェア取得部520で取得したソースコードの中に対象となるソースコードがあれば、被テストソフトウェア更新パッチ102のソースコードパッチ410を適用する(S532)。また、パッチを適用したソースコードとソフトウェア取得部520が取得したソースコードをコンパイルして、実行可能なバイナリ形式のコンポーネントを生成する(S533)。

【0066】

ここで生成したコンポーネントは、ユーザシステム110にインストールされているアプリケーション121、ミドルウェア122、OS123のコンポーネントである。

【0067】

次に、インストール部540は、図13に示すように、ソフトウェア取得部520が取得したアプリケーション121、ミドルウェア122、OS123のバイナリ形式のコンポーネントと、コンパイル部530が生成したバイナリ形式のコンポーネントを取得する(S541)。そして、取得したコンポーネントをプラットフォーム選択部510が選択したプラットフォーム571にインストールする(S542)。

【0068】

次に、パッチ適用部550は、図14に示すように、被テストソフトウェア更新パッチ102を取得し、バイナリコードパッチ420が存在すれば、インストール部540がインストールしたコンポーネントの中から、該当するコンポーネントに対してバイナリコー

ドパッチを適用する（S551）。

【0069】

ここまでの手順により、ユーザシステム110と同じ環境を持つシステムがテスト環境201に構築されることになる。

【0070】

次に、検証テスト実行部560は、図15に示すように、テストパターンを利用して検証テストを実施する（S561）。ここでテストパターンは、例えばユーザから必要なテストパターンを取得する場合もあれば、ベンダが自身で用意する場合もある。そして、テスト結果を判定し（S562）、テストに成功すれば、テストに成功したソフトウェア更新パッチをテスト成功ソフトウェア更新パッチ103としてパッチ送信部134に渡し（S563）、テストに失敗していれば、そのテスト結果を記録する（S564）。ここでテスト失敗とは、検証テスト中にシステムクラッシュする場合や、目標となる機能を達成できなかった場合である。

【0071】

そして、テスト結果の記録を参照して、再度被テストソフトウェア更新パッチ102が選択され、テストに成功するまで上記の手順が繰り返されることになる。

【0072】

〔実施形態2〕

以下、本発明に係る第二の実施形態を、図16を用いて説明する。

図16は、本発明に係る第二の実施形態のシステム更新方法を実現するためのシステム構成図である。

【0073】

第一の実施形態では、ベンダシステム131のテスト環境構築部133が、ベンダシステム133側にテスト環境を構築して、ベンダシステム133側でシステムのテストをおこなうものであった。

【0074】

本実施形態では、ユーザシステム110側にテスト環境を構築して、ベンダシステム側から、テストの対象となる被テストソフトウェア更新パッチ102を送信する例である。

【0075】

本実施形態のユーザシステムは、アプリケーション121と、ミドルウェア122と、OS123と、システム更新部111と、テスト環境構築部133aと、テスト環境201aとから構成されるシステムである。この内で、アプリケーション121と、ミドルウェア122と、OS123と、システム更新部111は、実施形態1と同様の構成である。

【0076】

また、ベンダシステム130は、ユーザシステム管理部131と、パッチ送信部134と、テスト結果フィードバック部135とから構成されるシステムである。

【0077】

ユーザシステムに構築されるテスト環境201aのプラットフォームは、ユーザのハードウェア環境とは、別に、実計算機を論理的に分割した論理計算機を用意して、その論理計算機にテスト環境201aを割り当てるようにする。これは、テストのためにユーザの使っている環境が影響を受けないようにするためである。

【0078】

本実施形態は、ユーザシステム110側に、テスト環境201aを構築するものである。そのため、ユーザシステム11からベンダシステム130にユーザシステム情報131が送付される。あるいは、ベンダシステム130側でユーザシステム情報を蓄積しておいてもよい。

【0079】

ベンダシステム130は、ユーザシステム110からシステム更新のリクエストがあると、パッチ送信部134からユーザシステム情報と被テストソフトウェア更新パッチ10

2を送付する。

【0080】

ユーザシステム110のテスト環境構築部133aでは、受け取ったユーザシステム情報と被テストソフトウェア更新パッチ102を元に、テスト環境を構築する。

【0081】

そして、テストが成功すると、その被テストソフトウェア更新パッチ102を元にして、システム更新機能111がユーザシステム110の更新をおこなう。

【0082】

テストが失敗すると、ユーザシステム110は、ベンダシステム130のテスト結果フィードバック部135に報告する。そして、テスト結果フィードバック部135は、次のテスト候補となる被テストソフトウェア更新パッチ102を再送付する。

【図面の簡単な説明】

【0083】

【図1】本発明に係る第一の実施形態のシステム更新方法を実現するためのシステム構成図である。

【図2】本発明に係る第一の実施形態のシステム更新方法の手順の概要を示すシーケンス図である。

【図3】ユーザシステム情報の一例を示す図である。

【図4】被テストソフトウェア更新パッチの一例を示す図である。

【図5】ユーザシステムの表示装置で、更新パッチの種別を選択する場面を示す模式図である。

【図6】ユーザシステムの表示装置で、更新テスト結果を確認し、更新を実行する場面を示す模式図である。

【図7】ユーザシステムのシステム更新の一形態を示す模式図である。

【図8】テスト環境構築部133の構成を示すブロック図である。

【図9】テスト環境201の構成例を示す模式図である。

【図10】テスト環境構築部133のプラットフォーム選択部510の動作を示すフローチャートである。

【図11】テスト環境構築部133のソフトウェア取得部520の動作を示すフローチャートである。

【図12】テスト環境構築部133のコンパイル部510の動作を示すフローチャートである。

【図13】テスト環境構築部133のインストール部520の動作を示すフローチャートである。

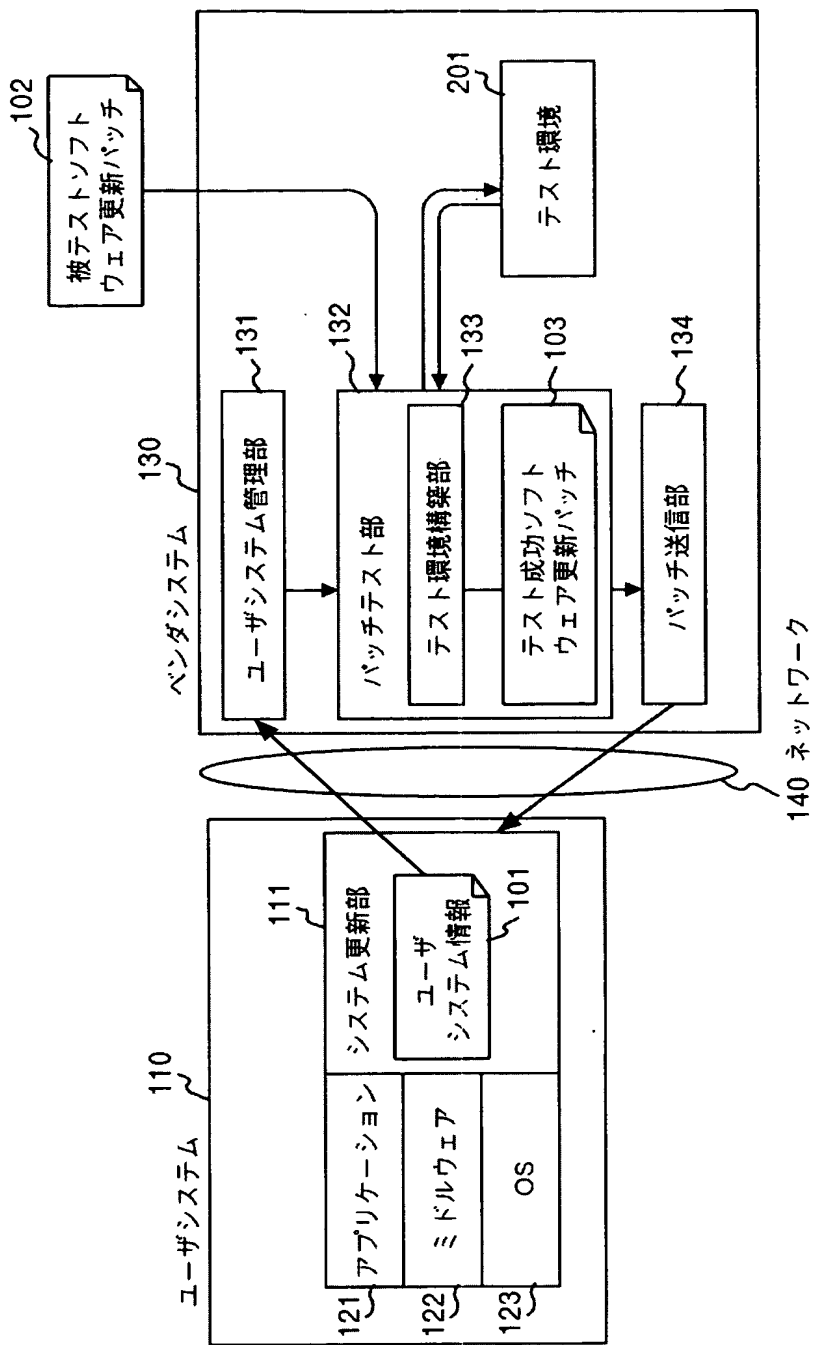
【図14】テスト環境構築部133のパッチ適用部550の動作を示すフローチャートである。

【図15】テスト環境構築部133の検証テスト実行部560の動作を示すフローチャートである。

【図16】本発明に係る第二の実施形態のシステム更新方法を実現するためのシステム構成図である。

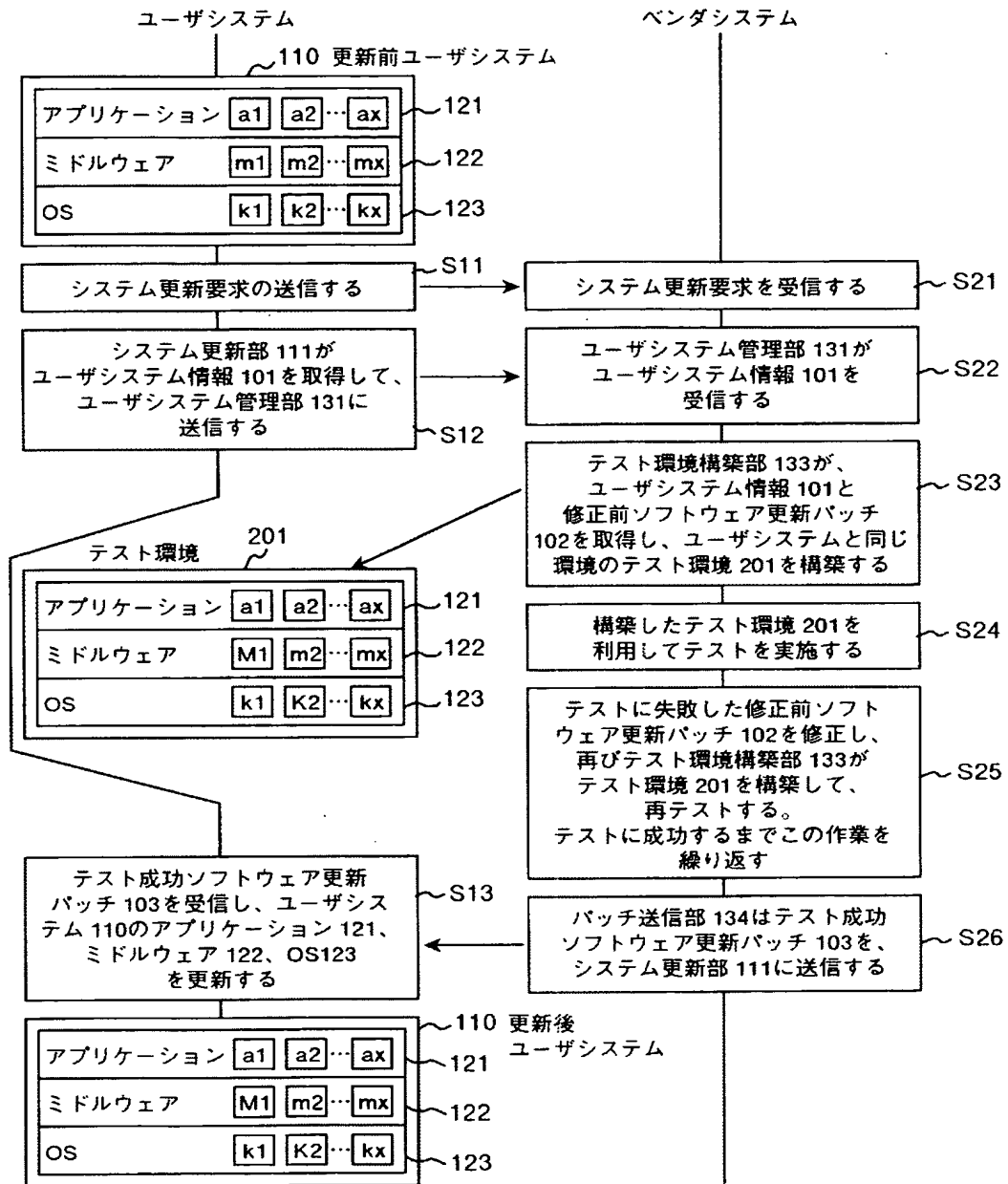
【書類名】 図面
【図1】

図 1



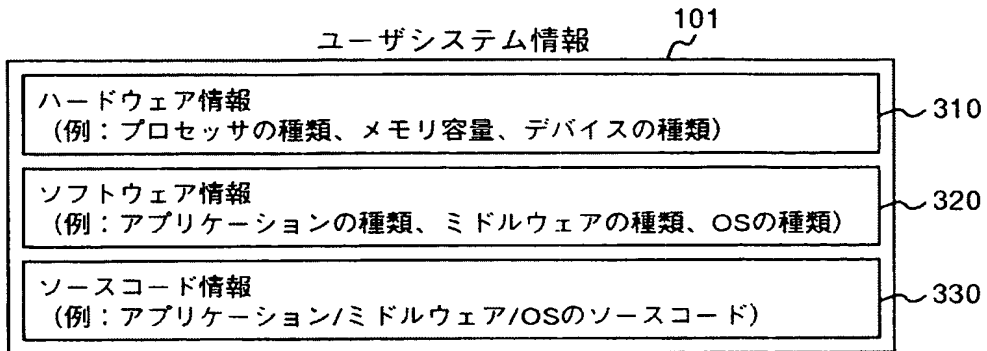
【図 2】

図 2



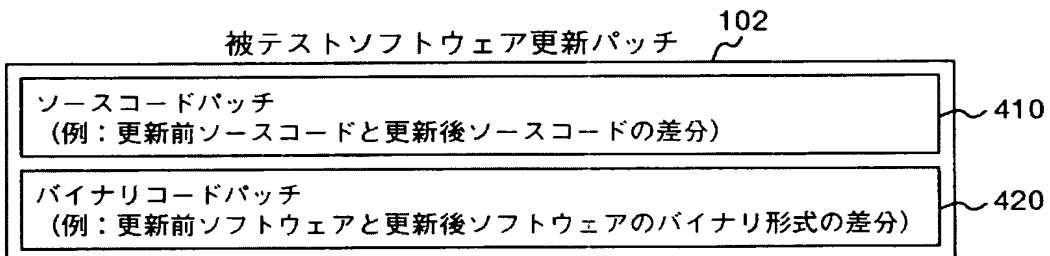
【図 3】

図 3



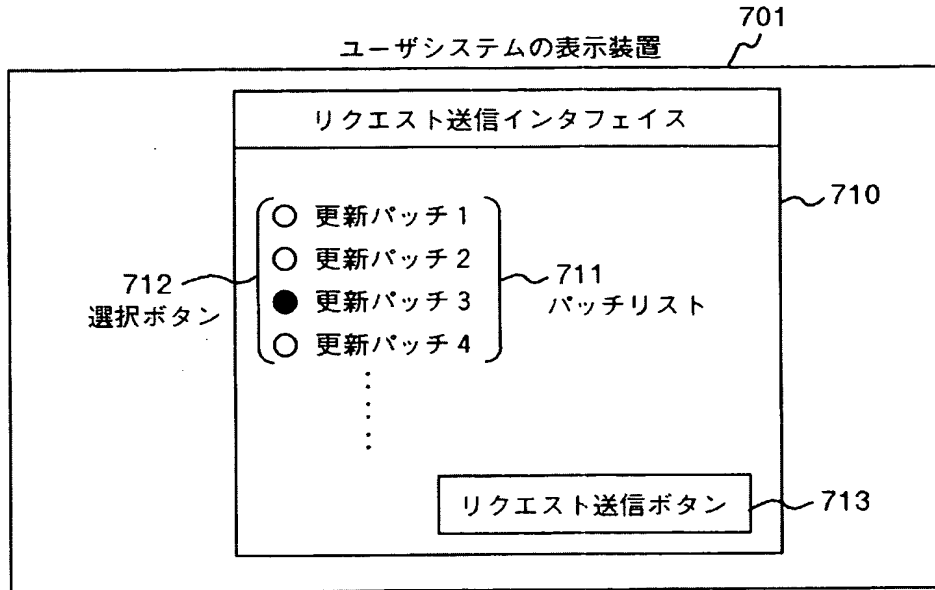
【図 4】

図 4



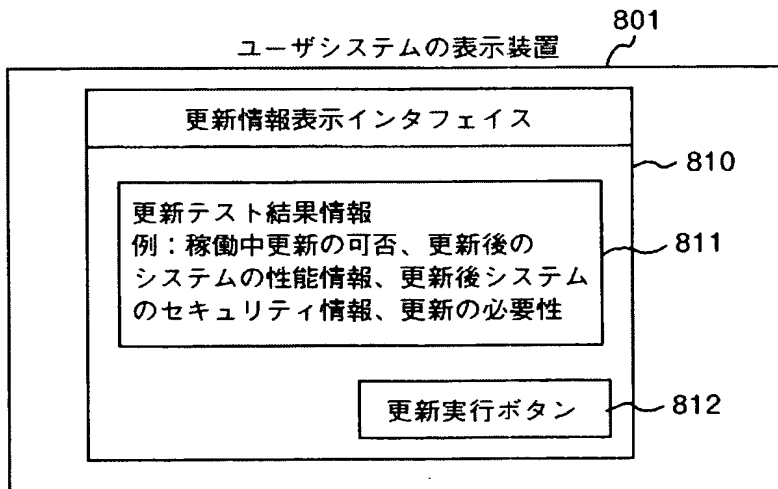
【図 5】

図 5



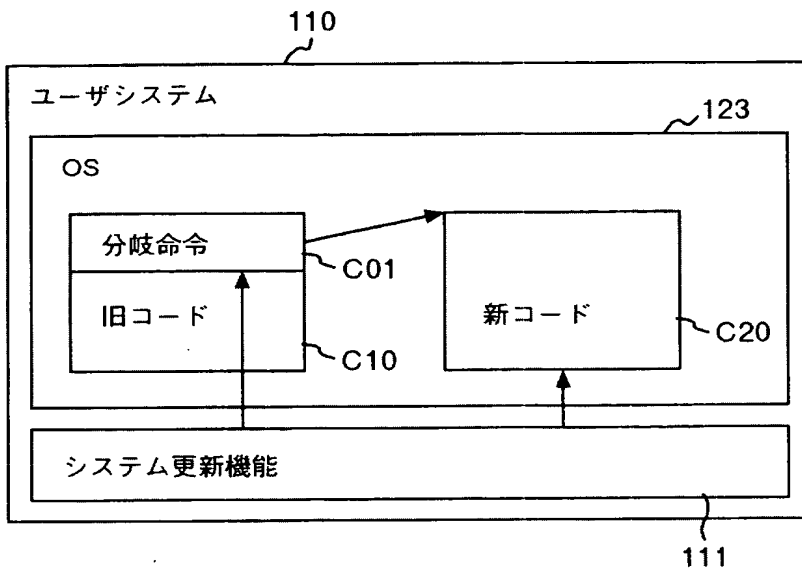
【図 6】

図 6



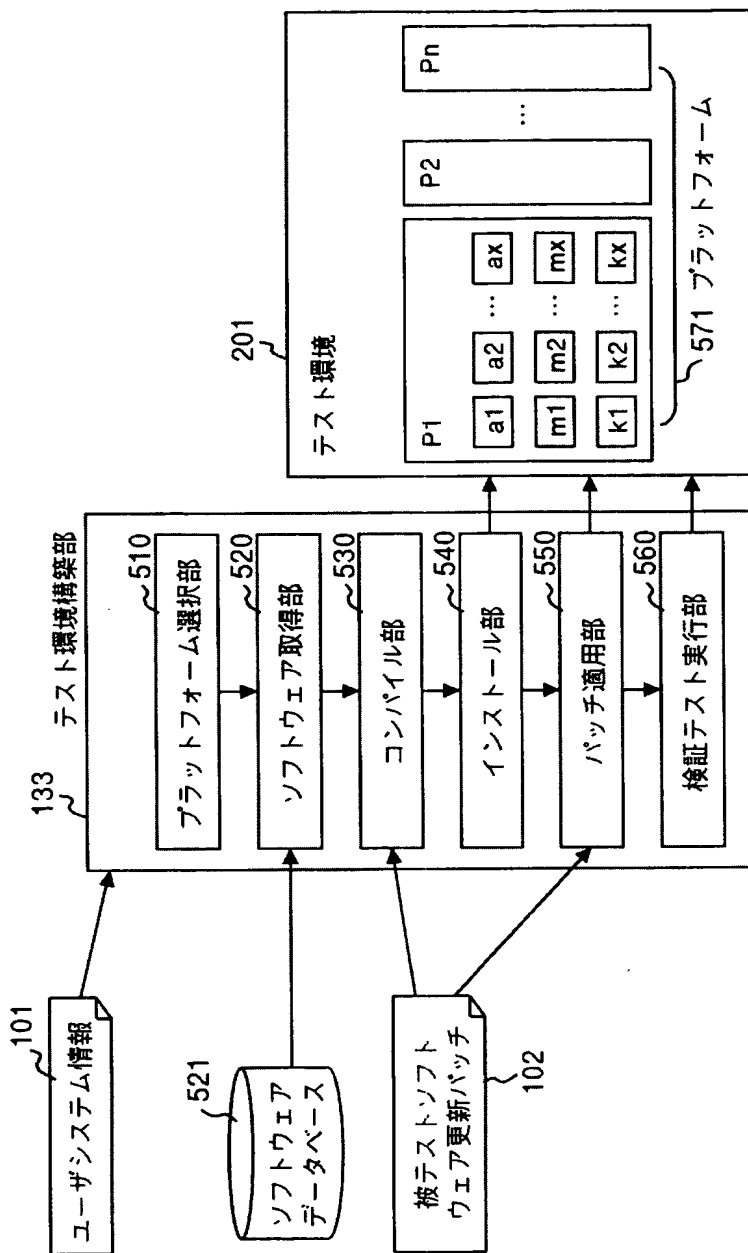
【図 7】

図 7



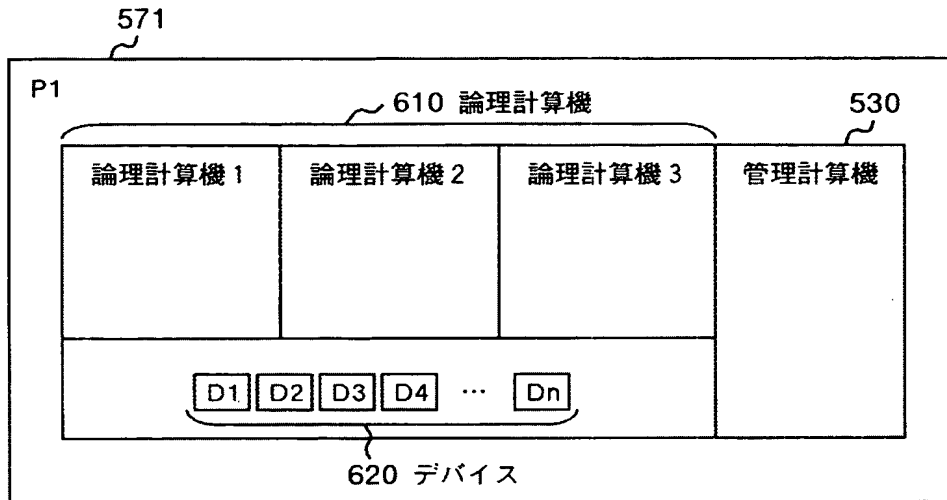
【図 8】

図 8



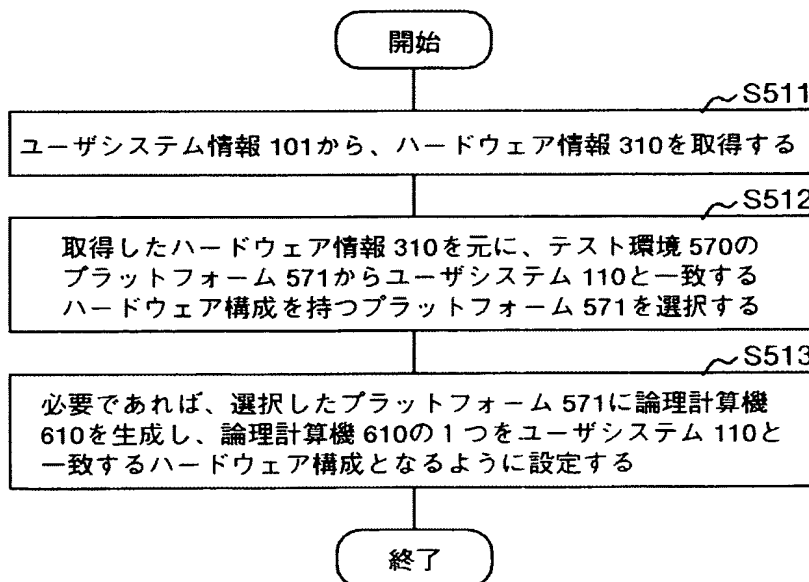
【図 9】

図 9



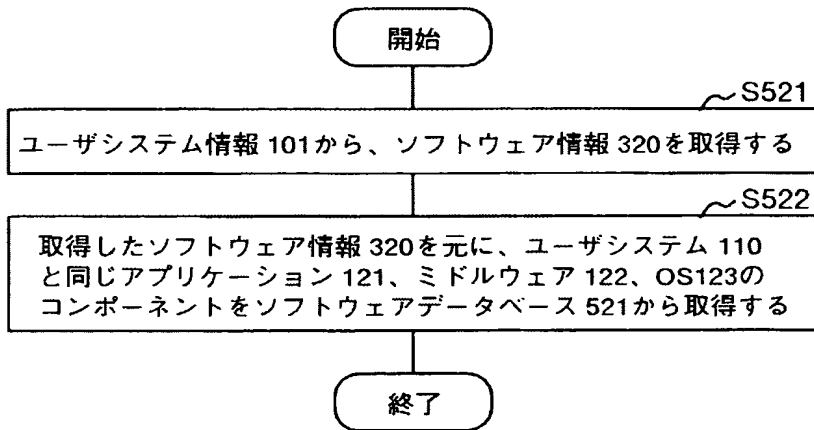
【図 10】

図 10



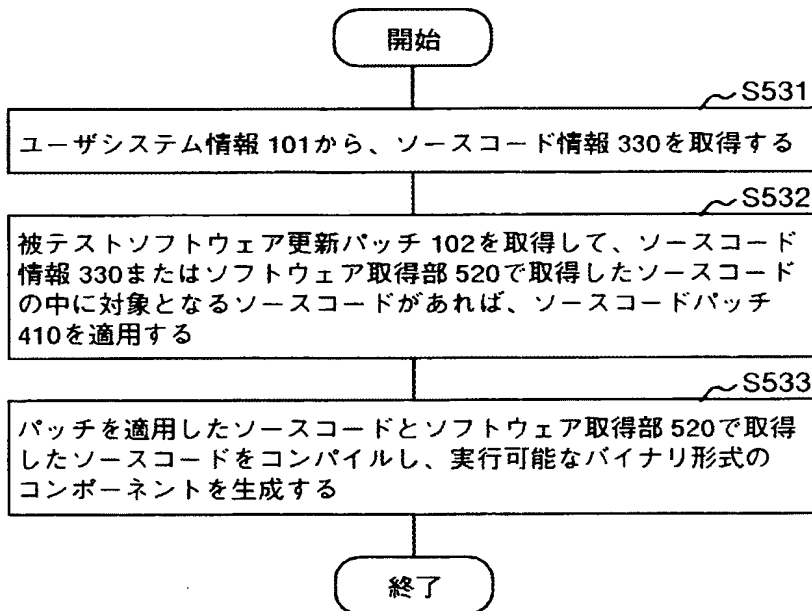
【図 11】

図 11



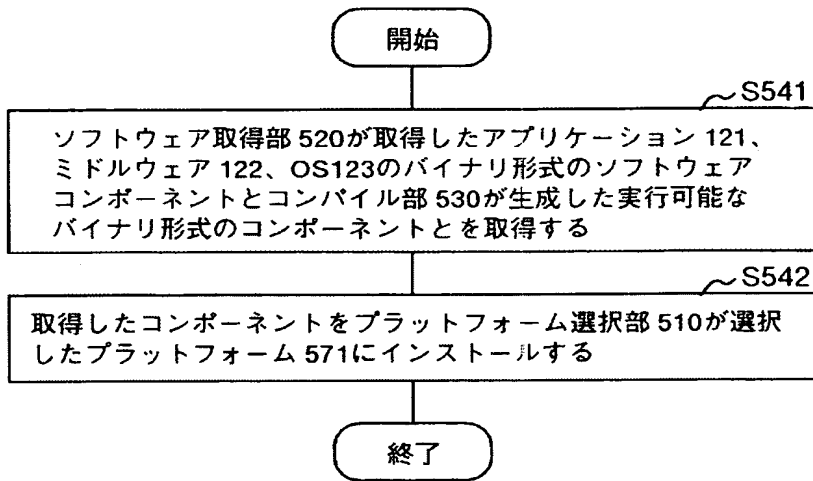
【図 12】

図 12



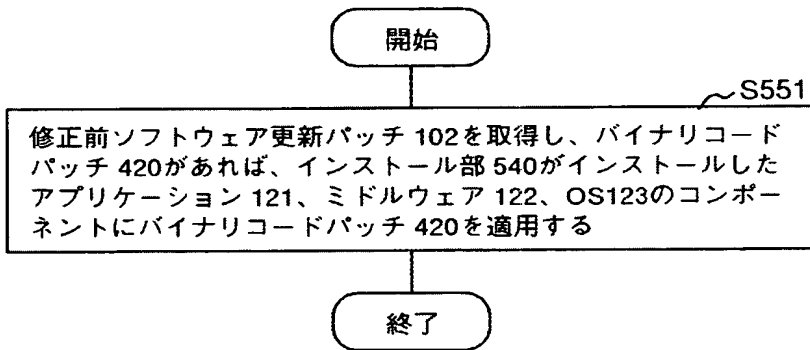
【図 13】

図 13



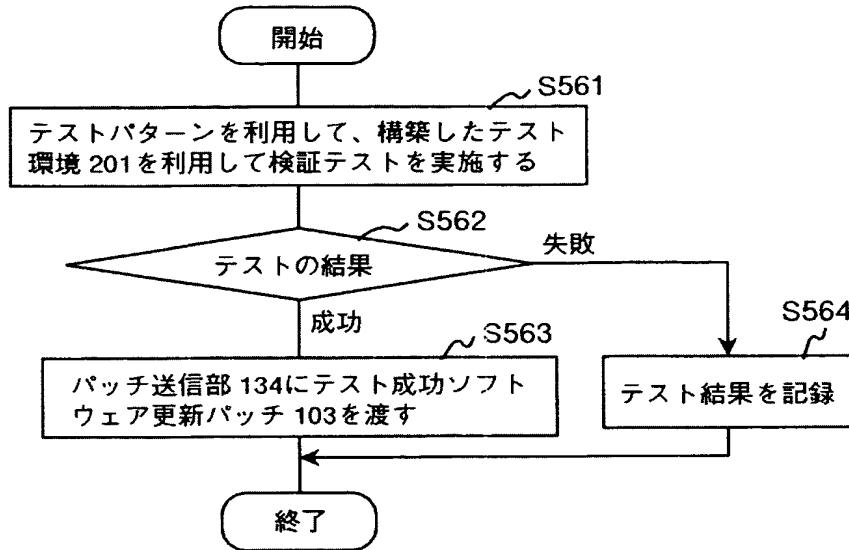
【図 14】

図 14



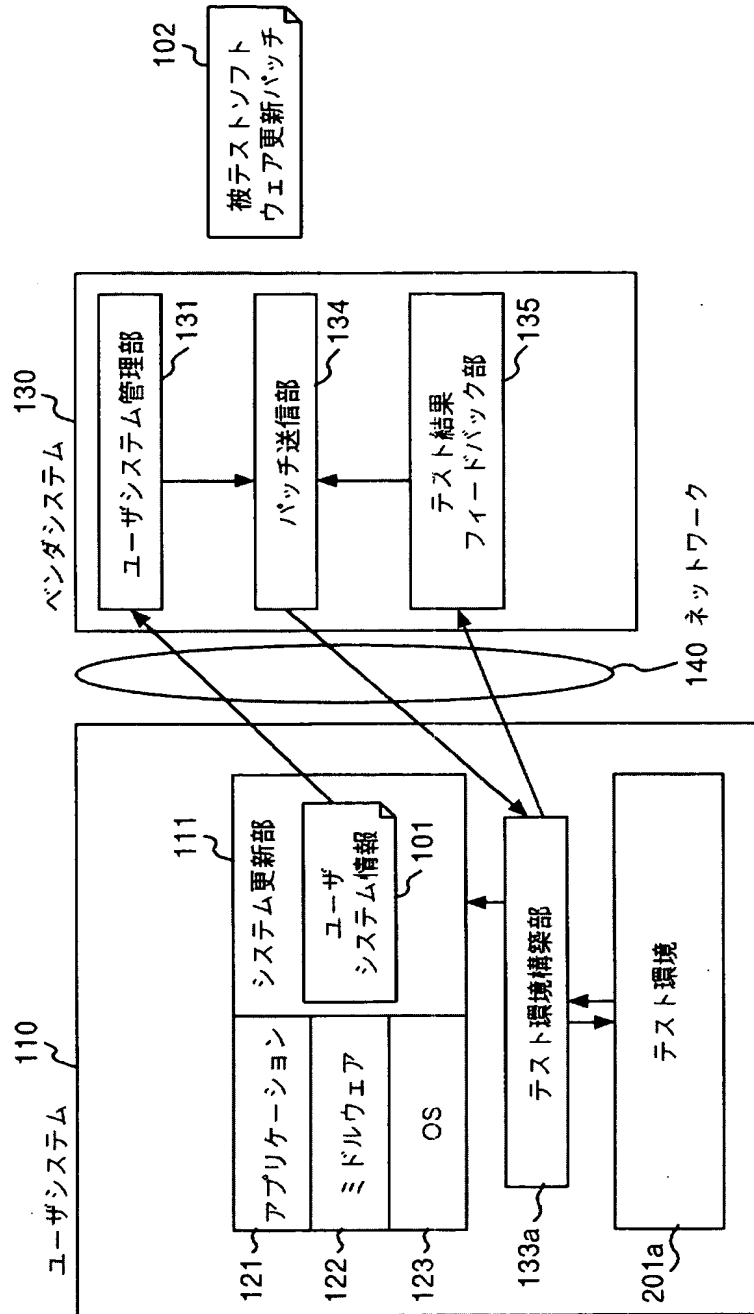
【図 15】

図 15



【図 16】

図 16



【書類名】 要約書**【要約】**

【課題】 ネットワークを介して、ソフトウェアを更新する場合に、システムの更新を簡単におこなえるようにし、しかも、ソフトウェア更新後のユーザシステムの信頼性とサービスレベルを保証する。

【解決手段】 ベンダシステムで、ユーザシステムにインストールされているソフトウェア情報やハードウェア情報を含むユーザシステム情報を取得する。そして取得したユーザシステム情報を基にテスト環境を構築して、そのテスト環境においてソフトウェア更新パッチを適用したシステムの動作を検証し、テスト環境においてシステムが正常に動作するかを確認する。そして、正常動作が確認された場合に、ソフトウェア更新パッチをユーザシステムに送信し、ユーザシステムでは、そのソフトウェア更新パッチを適用して、自らのシステムを更新する。

【選択図】 図 1

特願 2003-339487

出願人履歴情報

識別番号

[000005108]

1. 変更年月日

1990年 8月31日

[変更理由]

新規登録

住所

東京都千代田区神田駿河台4丁目6番地

氏名

株式会社日立製作所